

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ**

Кафедра системного програмування і спеціалізованих комп'ютерних систем

До захисту допущено

Завідувач кафедри

Віталій РОМАНКЕВИЧ

(підпис)

(ініціали, прізвище)

“     ”                      2020 р.

**Дипломний проєкт**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Системне програмування»**

спеціальності

**123 «Комп'ютерна інженерія»**

на тему: Студентська соціальна мережа для мобільних платформ на базі операційної системи Android

Виконав

студент IV курсу, групи КВ-61

Ціпун Владислав Юрійович

(прізвище, ім'я, по батькові)

(підпис)

Керівник доц. каф. СПіСКС, к. т. н., доцент Тарасенко-Клятченко О.В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант з нормоконтролю, доц.каф.СПСКС, к.т.н. Клятченко Я.М.

(назва розділу)      (посада, вчене звання, науковий ступінь, прізвище, ініціали)

(підпис)

Рецензент

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному  
проєкті немає запозичень з праць інших  
авторів без відповідних посилань.

Студент

(підпис)

Київ – 2020 року

**ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ**

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4		Завдання на дипломний проєкт	2	
2	A4	ІАЛЦ.045440.001 ОА	Опис альбому	2	
3	A4	ІАЛЦ.045440.002 ТЗ	Технічне завдання	4	
4	A4	ІАЛЦ.045440.003 ТП	Відомість технічного проєкту	2	
5	A4	ІАЛЦ.045440.004 ПЗ	Пояснювальна записка	44	
6	A4	ІАЛЦ.045440.005 Д1	Схема структурна	1	
7	A4	ІАЛЦ.045440.006 Д2	Схема алгоритму	1	
8	A4	ІАЛЦ.045440.007 Д3	Схема структурна	1	
9	A4	ІАЛЦ.045440.008 Д4	Схема алгоритму	1	

				ДП ІАЛЦ.045440.000		
	ПІБ	Підп.	Дата			
Розробн.	Олійник			Відомість дипломного проєкту	Лист	Листів
Керівн.	Тарасенко-Клятченко				1	1
Консульт.	Клятченко				КПП ім. Ігоря Сікорського Каф. СПіСКС Гр. KB-61	
Н/контр.	Клятченко					
Зав.каф.	Романкевич					

# **Пояснювальна записка до дипломного проєкту**

на тему: Студентська соціальна мережа для мобільних платформ на базі операційної системи Android

Київ – 2020 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ**

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Системне програмування»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Віталій РОМАНКЕВИЧ  
(підпис) (ініціали, прізвище)

«\_\_» \_\_\_\_\_ 2020 р.

**ЗАВДАННЯ  
на дипломний проєкт студента  
Ціпуна Владислава Юрійовича  
(прізвище, ім'я, по батькові)**

1. Тема проєкту «Студентська соціальна мережа для мобільних платформ на базі операційної системи Android»,  
керівник проєкту доц. каф. СПіСКС, к. т. н., доцент Тарасенко-Клятченко О.В.,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «\_\_» \_\_\_\_\_ 2020 р. № \_\_\_\_\_

2. Термін подання студентом проєкту

3. Вихідні дані до проєкту:

- програмне забезпечення що реалізує клієнт для внутрішньої соціальної мережі, орієнтованої на використання у навчальних закладах.

#### 4. Зміст пояснювальної записки

- аналіз основних особливостей використання соціальних мереж та обґрунтування теми дипломного проєкту;
- аналіз технологій для розробки сучасних клієнт-серверних додатків;
- технічна реалізація студентської соціальної мережі на базі операційної системи Android ;
- опис розробленого додатку.

#### 5. Перелік графічного матеріалу:

- структура програми (схема структурна);
- схема життєвого циклу фрагменту (схема алгоритму);
- розташування модулів програми (схема структурна);
- алгоритм роботи з FirebaseAuth (схема алгоритму).

#### 6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я.М., к.т.н., доцент каф. СПіСКС		

#### 7. Дата видачі завдання “30” жовтня 2019 р.

##### Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення літератури за тематикою проєкту	18.11.2019	
2.	Розроблення та узгодження технічного завдання	30.11.2019	
3.	Аналіз існуючих рішень	10.01.2020	
4.	Підготовка матеріалів першого розділу проєкту	18.01.2020	
5.	Розроблення програмного забезпечення	16.02.2020	
6.	Відлагодження програмного продукту	14.03.2020	
7.	Підготовка матеріалів другого розділу проєкту	01.04.2020	
8.	Розроблення інтерфейсу програмного забезпечення	20.04.2020	
9.	Підготовка графічної частини	01.05.2020	
10.	Оформлення документації дипломного проєкту	14.05.2020	

Студент

\_\_\_\_\_ (підпис)

Владислав ЦІПУН

Керівник проєкту

\_\_\_\_\_ Оксана ТАРАСЕНКО-КЛЯТЧЕНКО

(підпис)

## АНОТАЦІЯ

Кваліфікаційна робота включає пояснювальну записку (44 с., 19 рис., 4 додатки).

Об'єкт розробки – створення мобільного застосунку, який реалізує клієнт для соціальної мережі, що орієнтована на використання у закладах освіти.

Розроблене програмне забезпечення дозволяє:

- Користуватися повним функціоналом звичних нам соціальних мереж;
- Оптимізувати внутрішні процеси у навчальному закладі пов'язані з організацією комунікації зі студентами;
- Створити єдину базу знань (Wiki);
- Проводити опитування у зручний спосіб для налагодження співпраці з адміністрацією закладу.

В ході виконання дипломного проєкту:

- розроблено додаток для мобільних пристроїв на базі операційної системи Android;
- проведено аналіз існуючих рішень;
- розроблено структуру взаємодії клієнта та серверу.

Ключові слова: ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, СОЦІАЛЬНІ МЕРЕЖІ, ANDROID, АНАЛІТИКА, FIREBASE, JAVA, KOTLIN, WEB-ДОДАТОК, ХМАРНІ ТЕХНОЛОГІЇ, ВІДДАЛЕНИЙ СЕРВЕР, КЛІЄНТ.



## **ABSTRACT**

Qualification work includes an explanatory note (44 pages, 19 figures, 4 appendices).

The object of development is the creation of a mobile application that is implemented by the client for a social network focused on use in educational institutions.

The developed software allows:

- Use the full functionality of our usual social networks;
- Optimize the internal processes in the institution related to the organization of communication with students;
- Create a single knowledge base (Wiki);
- Conduct surveys in a convenient way to establish cooperation with the administration of the institution.

During the implementation of the diploma project:

- developed an application for mobile devices based on the Android operating system;
- the analysis of existing decisions is carried out;
- the structure of interaction of the client and the server is developed.

Keywords: SOFTWARE, SOCIAL NETWORKS, ANDROID, ANALYTICS, FIREBASE, JAVA, KOTLIN, WEB-APPLICATION, CLOUD TECHNOLOGIES.

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ. 045440.002 ТЗ	Студентська соціальна мережа для мобільних платформ на базі ОС Android	4		
			Технічне завдання			
	A4	ІАЛЦ. 045440.003 ТП	Студентська соціальна мережа для мобільних платформ на базі ОС Android	2		
			Відомість технічного проєкту			
	A4	ІАЛЦ. 045440.004 ПЗ	Студентська соціальна мережа для мобільних платформ на базі ОС Android	44		
			Пояснювальна записка			
	A4	ІАЛЦ. 045440.005 Д1	Структура програми	1		
			Схема структурна			

[illegible]

## ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ.....	2
2. ПІДСТАВА ДЛЯ РОЗРОБКИ .....	2
3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ .....	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	3
5. ТЕХНІЧНІ ВИМОГИ.....	3
5.1. Вимоги до програмного продукту, що розробляється .....	3
5.2. Вимоги до апаратного забезпечення.....	3
6. ЕТАПИ РОЗРОБКИ .....	4

					ІАЛЦ.045440.002 ТЗ										
Зм	Лист	№ докум.	Підп.	Дата	Студентська соціальна мережа для мобільних платформ на базі операційної системи Android <b>Технічне завдання</b>					Літ.	Аркуш	Аркушів			
Розроб.	Ціпун												1	4	
Перев.	Тарасенко -									НТУУ «КПІ ім. Ігоря Сікорського», ФПМ, КВ-61					
	Клятченко														
Н. контр.	Клятченко														
Затв.	Романкевич														

## **1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ**

Назва розробки: «Студентська соціальна мережа для мобільних платформ на база ОС Android».

Галузь застосування: розробка сучасних мобільних додатків на основі сучасних інформаційних технологій.

## **2. ПІДСТАВА ДЛЯ РОЗРОБКИ**

Підставою для розробки є завдання на виконання роботи першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

## **3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ**

Метою даного проєкту є створення програмного забезпечення для використання у закладах освіти для взаємодії викладацького штату зі студентами і зручного обміну інформацією.

					ІАЛЦ.045440.002 ТЗ	Арк
						2
Зм	Лист	№ докум.	Підп.	Дата		

#### **4. ДЖЕРЕЛА РОЗРОБКИ**

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації у періодичних виданнях та електронні статті у мережі Інтернет.

#### **5. ТЕХНІЧНІ ВИМОГИ**

##### **5.1. Вимоги до програмного продукту, що розробляється**

- Забезпечення необхідного функціоналу соціальної мережі у межах навчальних установ;
- Зручність кастомізації;
- Сповіщення про виняткові ситуації;
- Масштабована архітектура;
- Відмовостійкість;

##### **5.2. Вимоги до апаратного забезпечення**

- 15 МБ дискового простору на накопичувачі;
- Операційна система Android 4.0 Ice Cream Sandwich або більш сучасна версія.
- Наявність доступу до мережі Internet (Ethernet);

					ІАЛЦ.045440.002 ТЗ	Арк
						3
Зм	Лист	№ докум.	Підп.	Дата		

## 6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів
1.	Вивчення літератури за тематикою проєкту	15.04.2020
2.	Розроблення та узгодження технічного завдання	30.04.2020
3.	Аналіз існуючих рішень	05.05.2020
4.	Підготовка матеріалів першого розділу дипломного проєкту	10.05.2020
5.	Підготовка матеріалів другого розділу дипломного проєкту	18.05.2020
6.	Підготовка графічної частини дипломного проєкту	20.05.2020
7.	Оформлення документації дипломного проєкту	25.05.2020
8.	Попередній огляд матеріалів диплому на кафедрі	30.05.2020

[illegible]



[illegible]

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	3
ВСТУП	5
1. АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ ЗАСОБІВ ВНУТРІШНЬОЇ КОМУНІКАЦІЇ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ	6
1.1. Причини популярності та використання соціальних мереж	6
1.2. Корпоративні соціальні мережі. Аналіз	6
1.3. Портал та корпоративна соц мережа. Відмінності та особливості	7
1.4. Функціонал КСМ	9
1.5. Використання соціальних мереж у навчальних закладах	10
1.6. Аналіз платформ для комунікації зі студентами	11
1.7. Обґрунтування теми	12
2. АНАЛІЗ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ СУЧАСНИХ КЛІЄНТ-СЕРВЕРНИХ ДОДАТКІВ	13
2.1. Клієнт серверна архітектура	13
2.2. Сокети	17
2.3. Огляд Firebase	18
3. ТЕХНІЧНА РЕАЛІЗАЦІЯ СТУДЕНТСЬКОЇ СОЦІАЛЬНОЇ МЕРЕЖІ НА БАЗІ ОПЕРАЦІЙНОЇ СИСТЕМИ ANDROID	28
3.1. Вибір платформи для розробки ПЗ	28
3.2. Життєвий цикл розробки ПЗ	Ошибка! Закладка не определена.

					<b>ІАЛЦ.0454400.004 ПЗ</b>			
<b>Зм</b>	<b>Лист</b>	<b>№ докум.</b>	<b>Підп.</b>	<b>Дата</b>	<b>Тема</b> Студентська соціальна мережа для мобільних платформ на базі операційної системи Android <b>Пояснювальна записка</b>	<b>Лім.</b>	<b>Лист</b>	<b>Листів</b>
<b>Розроб.</b>	Ціпун						1	48
<b>Перев.</b>	Тарасенко-					<b>КПІ ім. І. Сікорського, ФПМ, КВ-61</b>		
	Клятченко							
<b>Н. контр.</b>	Клятченко							
<b>Затв.</b>	Романкевич							

3.3. Вибір інструментарію _____	35
3.4. Опис архітектури додатку _____	37
4. ОПИС РОЗРОБЛЕНОГО ДОДАТКУ _____	38
4.1. Опис інтерфейсу додатку _____	38
4.2. Рекомендації щодо покращення додатку _____	41
ВИСНОВКИ _____	42
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ _____	43

## ДОДАТКИ

### Додаток А. Копії графічних матеріалів

- ІАЛЦ.045440.005 Д1. Структура програми. Схема структурна;
- ІАЛЦ.045440.006 Д2. Схема життєвого циклу фрагменту. Схема алгоритму;
- ІАЛЦ.045440.007 Д3. Розташування модулів. Схема структурна;
- ІАЛЦ.045440.008 Д4. Алгоритм роботи з FirebaseAuth. Схема алгоритму.

### Додаток Б. Лістинг програми

### Додаток В. Презентація бакалаврського проєкту

## ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

JSON (JavaScript Object Notation) – текстовий формат обміну даних, оснований на JavaScript.

KCM – корпоративна соціальна мережа.

ПЗ – програмне забезпечення.

TCP (Transport Control Protocol) – протокол транспортного рівня передачі даних в моделі OSI.

HTTP (Hyper Text Transfer Protocol) – протокол прикладного рівня передачі даних в моделі OSI.

ECM (Enterprise Content Managment) – управління корпоративним контентом.

IP (Internet Protocol) – протокол мережевого рівня для передавання даних між мережами, основа стеку протоколів TCP/IP.

FTP (File Transfer Protocol) – протокол призначений для передачі файлів в комп'ютерних мережах.

OSI – абстрактна мережева модель для комунікації і розробки мережевих протоколів.

СУБД- система управління базами даних.

Інtranет – внутрішньокорпоративна мережа, що використовує стандартні технології і програмне забезпечення інтернету.

BAAS (Backend-as-a-Service) – модель, що дозволяє розробникам веб-додатків і мобільних застосунків зв'язати їх додатки з серверним хмарним сховищем та API.

Деплой – розгортання програмного забезпечення.

AMPPS- набір рішень, що включає в себе Apache, MySQL, MongoDB, PHP, Python.

					<b>ІАЛЦ.0454400.004 ПЗ</b>	<b>Лист</b> 3
<b>Зм</b>	<b>Лист</b>	<b>№ докум.</b>	<b>Підп.</b>	<b>Дата</b>		

API (Application Programming Interface) – набір готових класів, процедур, функцій, структур і констант, що надаються додатком для використання в зовнішніх програмних продуктах.

SDK (Software Development Kit) – набір засобів розробки, утиліт і документації, який дозволяє програмістам створювати прикладні програми.

					<b><i>ІАЛЦ.0454400.004 ПЗ</i></b>	<b><i>Лист</i></b>
<b><i>Зм</i></b>	<b><i>Лист</i></b>	<b><i>№ докум.</i></b>	<b><i>Підп.</i></b>	<b><i>Дата</i></b>		<b><i>4</i></b>

## ВСТУП

На сьогоднішній день соціальні мережі стали невід'ємною частиною нашого життя, і з цим важко не погодитись, адже ключовим аспектом життя є спілкування та активна взаємодія між людьми, не зважаючи на відстань. Сучасні соціальні мережі можуть допомогти у вирішенні проблем не лише пов'язаних з пошуком однодумців чи товаришів, але й відіграють велику роль у ціленапрявленому вирішенні професіональних питань, оптимізації введення бізнесу, чи спрощення організації навчального процесу.

Корпоративна соціальна мережа – це певний аналог звичайної соціальної мережі, але адаптована для вирішення професійних потреб та спрощення організації внутрішніх процесів діяльності організації.

Ці рішення станом на сьогодні не набули ще великої популярності як у сфері ведення звичайного бізнесу, так і у навчальних закладах, де питання організації взаємодії викладацького штату зі студентами стає дедалі гострішим з кожним роком. У 2020 році навчальні заклади на практиці зіткнулися з проблемами комунікації студентів з викладачами під час дистанційного навчання. Листування електронною поштою у таких випадках є не самим зручним способом, оскільки не всі користувачі мають доступ до Університетських локальних ресурсів.

Активному поширенню таких засобів заважає певна складність інтеграції подібних рішень, а також безпосередня консервативність керівного складу, котрий бачить використання подібних додатків у більшості випадків як потенційного винуватця неефективності праці організації.

За думкою багатьох відомих експертів, розповсюдження КСМ у недалекому майбутньому та їх використання буде більш розповсюджене у всіх сферах, проте необхідний аналіз можливого використання КСМ у навчальних закладах.

					<b>ІАЛЦ.0454400.004 ПЗ</b>	Лист
						5
Зм	Лист	№ докум.	Підп.	Дата		

# 1. АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ ЗАСОБІВ ВНУТРІШНЬОЇ КОМУНІКАЦІЇ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ

## 1.1 Причини популярності та використання соціальних мереж

Протягом останніх десятиліть людство швидко розвивається та знаходить ще більше нових можливостей для покращення та спрощення свого життя, і це стосується усіх сфер діяльності. Не виключенням є і спілкування. Використання сучасних технологій дозволило людству залишатися на зв'язку з будь-якої точки земної кулі. Завдяки розвитку комп'ютерних технологій стало можливим створення єдиного середовища, основою якого є використання комп'ютерних мереж та засобів зв'язку.

Соціальними мережами називають спеціалізовані онлайн платформи, що дозволяють користувачам вести спілкування, обмінюватись різноматним медіа контентом, залишатися в курсі останніх новин, об'єднуватись у групи за інтересами. Сучасні мобільні засоби зв'язку значно спрощують використання подібних програмних засобів, адже дозволяють мати під рукою повнофункціональний пристрій, що може легко замінити звичайний стаціонарний персональний комп'ютер для вирішення повсякденних задач.

## 1.2 Корпоративні соціальні мережі. Аналіз

Проте є хибним ствердження того, що соціальні мережі призначені лише для розваг. Існує багато прикладів так званих професійних соціальних мереж, що створюються задля об'єднання користувачів для обміну досвідом, спілкуванням за професійною тематикою, розміщенням вакансій, розвитку ділових взаємовідносин, наприклад таких як LinkedIn.

Не менш важливими у професійній діяльності є корпоративні соціальні мережі. Корпоративна соціальна мережа – це певний аналог звичайної соціальної мережі, але адаптована для вирішення професійних потреб та спрощення організації внутрішніх процесів діяльності організації. Вони з'явилися відносно нещодавно, на фоні росту популярності звичайних соціальних мереж. Варто розуміти різницю між корпоративною соціальною мережею та інтранетом.

Інтранет – це внутрішня мережа організації, обмежена віртуальним простором окремо взятої організації, захищений від проникнення на корпоративні вузли.

Інтранет зовсім не є аналогом корпоративної соціальної мережі. Зазвичай під цим поняттям мають на увазі певний внутрішньокорпоративний сайт, або навіть декілька сайтів. Виділяють дві технології його побудови:

- 1) ЕСМ – системи;
- 2) Корпоративні портали;

Тобто, в порівнянні з соціальною мережею – це більш широке поняття, що об'єднує кілька різних складових. Проте на практиці під Інтранетом розуміють корпоративний портал. Нажаль, нерідко такі портали не користуються попитом у користувачів, та в більшості випадків програють месенджерам та соціальним мережам.

### 1.3 Портал та корпоративна соціальна мережа. Відмінності та особливості.

Перш за все, портал має особливість, яка полягає у тому, що він не передбачає вільного спілкування між користувачами. В основному – це спеціальне середовище для зберігання офіційної інформації структурного підрозділу, що володіє ним. Психологія людини побудована таким чином, що



вона ніколи не надасть перевагу офіційним робочим інструментам для вільного спілкування. Це відноситься, наприклад, до електронної пошти.

По-друге, портал потребує наявності у штаті адміністратора. Саме адміністратор наповнює його контентом, створює спільноти для обговорення, надає доступ користувачам до внутрішніх файлів. Ця властивість робить портали не мобільними, позбавленими можливості оперативно реагувати на вимоги та потреби користувачів. Наприклад, рядовий співробітник є позбавленим права терміново змінювати контент порталу, а саме: розміщувати важливі оголошення, ділитися власними матеріалами тощо.

Всі ці особливості роблять портал не надто популярним серед користувачів, незважаючи на його користь у певних ситуаціях.

У корпоративних соціальних мережах ці процеси відбуваються навпаки: контент створюється безпосередньо користувачами. Будь-який співробітник має можливість для створення обговорення питань, мають доступ до безпосередньої бази знань (wiki), що може бути створена на основі набутого досвіду колег. Ця система бази знань є дуже універсальною та корисною для рішення практичних задач, адже досить лише скористатися пошуковим рядком, записавши туди ключові слова для пошуку, і отримати результат.

Також безперечною перевагою є те, що інтерфейс та знайомий функціонал роблять корпоративні соціальні мережі дуже зручними та простими у використанні, адже нема необхідності у витраченні часу на знайомство з системою та звикання до неї.

КСМ формує в групі певну інформаційну спільноту, з розвиненими горизонтальними та вертикальними зв'язками, наприклад кожен користувач може взаємодіяти не лише один з одним, але й з керівним штатом. Такі мережі особливо корисними є у ситуаціях, коли треба об'єднувати організації з великим штатом та кількістю філіалів у різних районах. Мережа забезпечує доступ до повного списку контактів користувачів, знайти потрібну людину можна не лише за допомогою прізвища та імені, але й посади чи підрозділу, де

					<i><b>ІАЛЦ.0454400.004 ПЗ</b></i>	<i>Лист</i> 8
<i>Зм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		

вона знаходиться. Також мережа стає інформаційною базою, до якої можна звертатися кожному бажаючому, а відповідно скорочуються витрати часу на пошук потрібної інформації, забезпечуються оптимальні взаємозв'язки між користувачами. За допомогою інтегрованих чатів можна зручно організовувати обговорення певних питань, брати участь у обговореннях.

Такі платформи можуть забезпечувати можливість для створення онлайн конференцій та семінарів, що у деяких ситуаціях є необхідною альтернативою очним зустрічам, проводити які не завжди доречно. Не варто також забувати про спрощену адаптацію новачків у колективі. КСМ допомагають вирішувати і питання соціалізації всередині колективу. На території таких мереж новачки швидше знаходять однодумців, нові знайомства, нових наставників, котрим можуть адресувати цікавлячі їх запитання. Як наслідок – налагодження комунікації всередині колективу, що значно спрощує взаємовідносини між ними.

#### 1.4 Функціонал КСМ.

Насправді, КСМ поєднує в собі всі можливості звичайних соціальних мереж, що також є перевагою перед згаданими порталами: користувачі звикли до їх функціонування, тому процес переходу на КСМ не буде приносити труднощі. Серед можливостей користувача – створення особистої сторінки, блогу, спільнот, груп, створення файлових баз, обмін між ними, стеження за визначеними подіями тощо. Необхідний аналіз базового функціоналу корпоративних соціальних мереж:

Створення та обмін контентом виконується самостійно кожним з користувачів, причому обмеження відсутні, у більшості випадків. Це можуть бути як і текстові файли, так і фото та відеофрагменти, що зберігатимуться у внутрішній базі даних системи. Дуже важливою частиною мережі є реалізація функціоналу широкомовних сповіщень, що дозволяє доносити важливу

інформацію безпосередньо до кожного учасника групи/мережі (якщо мається на увазі більш глобальний рівень). Критичним є забезпечення різноманітних інструментів для проведення обговорень, таких як чати, відгуки, коментарі, блоги. Корисно мати так звану інформаційну стрічку, що містить останні новини організації, а також особистий/груповий календар, що може сповіщувати про заплановані події та активності, назначені самими учасниками чи керівним штатом.

### 1.5 Використання соціальних мереж у навчальних закладах.

Незважаючи на те, що КСМ на практиці є досить корисним інструментом, їх використання у навчальних закладах не набуло високого попиту станом на сьогодні. Керівний штат, зазвичай, сприймає ці платформи як засіб, який негативно впливає на навчальний процес, призводить до непродуктивної роботи, та і загалом відволікає від навчальних процесів. Проте позитивна роль подібних платформ є недооціненою. Сучасний цифровий світ сприяє масовій діджиталізації суспільства, що беззаперечно може впливати і на навчальні заклади, де соціальні мережі можуть вивести взаємодію викладачів та студентів на новий, більш сучасний та зручний рівень.

При правильному використанні, подібні платформи дійсно можуть бути корисними з точки зору навчальних процесів. Завдяки їм викладачі можуть легко зв'язуватись зі студентами та включати соціальні медіа в хід проведення лекцій, за необхідністю роблячи їх більш привабливими. Соціальні мережі – чудовий інструмент для взаємодії зі студентами, оскільки вони з ними добре знайомі, а також часто використовують подібні засоби у реальному повсякденному житті. Також допустиме використання додатку як інструмента для аналітики, досліджень, моніторингу. Організація адміністрацією опитувань щодо якості аудиторій та інтересу до предметів стане більш зручною, що дасть можливість проводити аналіз швидко та часто. Більш формальний спосіб

					<b>ІАЛЦ.0454400.004 ПЗ</b>	<b>Лист</b> 10
<b>Зм</b>	<b>Лист</b>	<b>№ докум.</b>	<b>Підп.</b>	<b>Дата</b>		

спілкування між викладачами та студентами сприяє покращенню стосунків викладач – студент. Хибно вважати, що звичайні соціальні мережі не використовуються для таких цілей, проте це не є зручним та правильним рішенням, адже доводиться використовувати особисті сторінки, використовувати які в робочих цілях не є хорошою практикою. Навчальні посібники, вказівки, матеріали не завжди передбачають доступ до них поза екосистемою навчального закладу. Саме таку проблему і призначені вирішувати спеціальні соціальні мережі, база даних якої знаходиться безпосередньо у межах комп'ютерної інфраструктури організації, і, відповідно, гарантується первинний доступ лише визначеним користувачам.

#### 1.6 Аналіз платформ для комунікації зі студентами.

Необхідно розглянути існуючі засоби для внутрішньої комунікації між співробітниками та студентами навчальних закладів на прикладі платформ, що використовуються у НТУУ «КПІ». Більшість факультетів використовують аналоги корпоративних порталів. Головним недоліком такого підходу є великий обсяг робіт при незначних модифікаціях, адмініструванні. Маніпуляції з вмістом portalу виконуються безпосередньо адміністратором сервісу або відповідальними особами. Як наслідок – витрата значного часу на оновлення, зміни, редагування контенту, односторонність у публікаціях, адже, зазвичай, контент створюється керівним штатом, тому виникає проблема неможливості децентралізації та постійна потреба в адміністраторах. В цілому портали не передбачають організацію комунікацій між студентами та відповідальними особами, а отже виникає потреба у використанні сторонніх сервісів. Ці недоліки є суттєвими у сфері використання сучасних технологій, виникає потреба в оптимізації внутрішніх процесів навчального закладу, підвищення продуктивності, розширення базового функціоналу portalу. Існуючі портали, в більшості, не передбачають реалізацію потрібного функціоналу, тому виникає

гостре питання стосовно заміни існуючого сервісу на більш функціональні зразки, що будуть мобільними та гнучкими у кастомізації, матимуть просту систему модерації, децентралізованість.

### 1.7 Обґрунтування теми.

Аналізуючи думки провідних експертів в області комп'ютерних технологій, можна стверджувати, що соціальні мережі для навчальних установ зможуть у близькому майбутньому здобути високу популярність, адже переваги цього сервісу дійсно іноді є дуже суттєвими. Проте на практиці неможливо обрати застосунок конкретно під ці потреби. Існуючі гіганти серед представників соціальних мереж, такі як Facebook чи WhatsUp не можуть бути цілковито інтегровані у інфраструктуру організації, оскільки вони є незалежними продуктами, і обмеження користувачів в цих застосунках реалізовано лише на рівні створення спільнот, що, звичайно, не є дуже зручним варіантом. У даному дипломному проєкті буде представлено повноцінний самостійний додаток, розрахований на використання у навчальних установах, гнучкий у кастомізації та інтеграції, орієнтований на взаємодію викладацького штату зі студентами.

## ВИСНОВОК ДО РОЗДІЛУ 1

Було проаналізовано ринок для обраної предметної області. Це надає можливість для більш конкретного розуміння особливостей використання корпоративних соціальних мереж та їх перспективи інтеграції до навчальних закладів.

Оскільки подібні програмні засоби ще не користуються великою популярністю, необхідно розвивати ідеї реалізації таких сервісів, адже створення конкурентоспроможного додатку може стати гарним стартом для більш масштабного розвитку спеціалізованих соціальних мереж. Порівняльний аналіз з існуючими аналогами дає можливість зрозуміти перспективи цих рішень.

					<b><i>ІАЛЦ.0454400.004 ПЗ</i></b>	<b><i>Лист</i></b>
						<b><i>13</i></b>
<b><i>Зм</i></b>	<b><i>Лист</i></b>	<b><i>№ докум.</i></b>	<b><i>Підп.</i></b>	<b><i>Дата</i></b>		

## 2. АНАЛІЗ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ СУЧАСНИХ КЛІЄНТ-СЕРВЕРНИХ ДОДАТКІВ

### 2.1 Клієнт серверна архітектура

Мільйони користувачів щодня користуються мережею Інтернет для того, щоб дізнатися останні новини, отримати корисну інформацію, обмінятися нею з іншими користувачами. Концепція, що дозволяє виконувати всі ці дії в мережі Інтернет отримала назву «клієнт-сервер». Клієнт виступає споживачем, а сервер – постачальником послуг. Сервером можуть бути наступні програми: всі HTTP сервера (Apache), MySQL сервер, локальний сервер AMPPS.

Клієнт та сервер взаємодіють між собою в мережі Інтернет, або в будь-якій іншій комп'ютерній мережі за допомогою різних мережевих протоколів, наприклад, IP протокол, HTTP протокол, FTP та інші. Кожен протокол дозволяє виконувати різні задачі. Наприклад, за допомогою HTTP протоколу клієнт може відправляти спеціальні HTTP повідомлення, в яких вказано, яку інформацію і в якому вигляді клієнт хоче отримати від сервера. Сервер, отримавши таке повідомлення, відправляє клієнту відповідь у вигляді повідомлення ( або декілька повідомлень) , схоже за структурою.

Повідомлення, які відправляє клієнтська частина, отримали назву HTTP запити. Запити мають спеціальні методи, котрі повідомляють серверу те, як необхідно оброблювати це повідомлення. Повідомлення, які відправляє сервер отримали, назву HTTP відповіді, вони містять в собі окрім корисної інформації також і спеціальні коди стану, котрі дозволяють клієнту визначити те, як сервер проаналізував та зрозумів його запит. Описані схематично дії відбуваються на сьомому рівні моделі OSI, але насправді ця взаємодія проходить на всіх семи рівнях. Коли клієнт відправляє запит, повідомлення упаковується, а коли повідомлення отримується сервером, то ці обгортки розгортаються. Також

варто відмітити те, що в основі взаємодії клієнт-сервер покладено принцип того, що цю взаємодію починає клієнт, а сервер – відповідає клієнту і повідомляє про те, чи може він надати якусь послугу клієнту, і якщо може, то на яких умовах.

Клієнтське програмне забезпечення і серверне програмне забезпечення зазвичай встановлюється на різних машинах, проте вони так само можуть працювати на одній.

Ця концепція взаємодії була розроблена в першу чергу для того, щоб розподілити навантаження між учасниками процесу обміну інформацією, а також для того, щоб розділити програмний код постачальника і споживача.

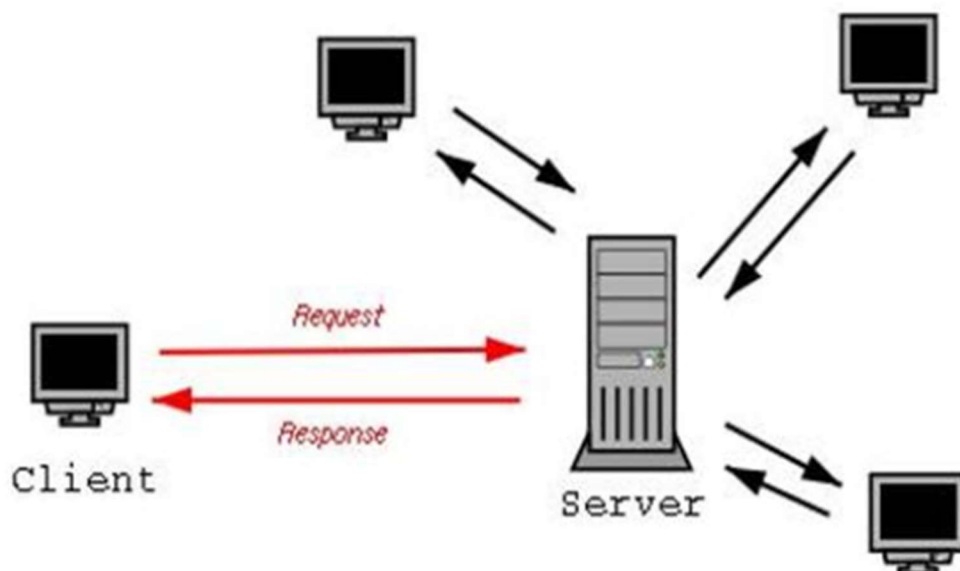


Рисунок 2.1 – Клієнт-серверна архітектура

Схема наглядно демонструє те, що до сервера може одночасно звертатися одразу декілька клієнтів. Також варто відмітити факт того, що кількість клієнтів, котрі можуть одночасно взаємодіяти з сервером, прямо залежить від потужності сервера та від того, який результат хоче отримати клієнт від сервера.

Багато мережових протоколів побудовані на архітектурі клієнт-сервер, тому в їх основі зазвичай покладені однакові або схожі принципи взаємодії, а



різниця помітна лише в деталях, котрі обумовлені особливостями та специфікою області, для якої розроблявся той чи інший мережевий протокол.

Архітектура клієнт-сервер визначає лише загальні принципи взаємодії між двома пристроями, деталі взаємодії яких визначаються мережевими протоколами. Ця концепція визначає те, що потрібно розділяти машини в мережі на клієнтські, котрі завжди посилають запити, та серверні, котрі віддають потрібні дані. При цьому взаємодію завжди починає клієнт, а правила, за якими відбувається взаємодія, визначається протоколами.

Існують два види архітектури взаємодії клієнт-сервер:

- 1) Двохрівнева архітектура клієнт-серверної взаємодії
- 2) Багаторівнева архітектура клієнт-серверної взаємодії (іноді називають трьохрівневою архітектурою)

Принцип роботи двохрівневої архітектури взаємодії клієнт-сервер полягає в тому, що обробка запита відбувається на одній машині без використання сторонніх засобів. Вона потребує високих системних вимог до сервера, але в той же час є дуже надійною.

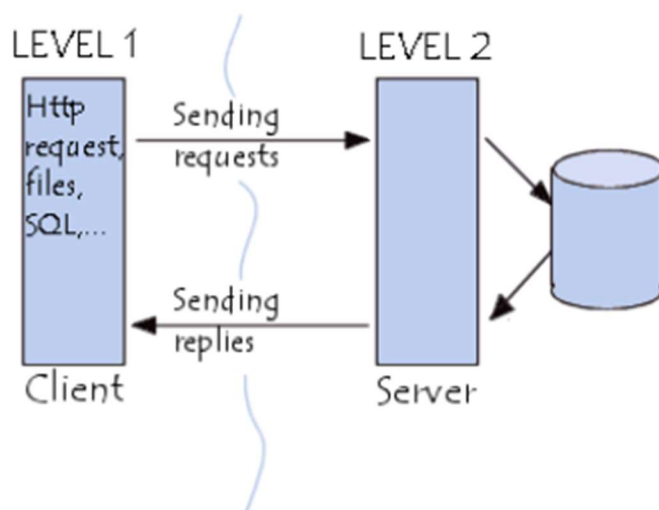


Рисунок 2.2 – Однорівнева клієнт-серверна архітектура

Схема демонструє те, що існує клієнт (1-ий рівень), котрий дозволяє користувачеві зробити запит, і є сервер, котрий обробляє запит клієнта.

Суть багаторівневої архітектури полягає в тому, що запит клієнта оброблюється одразу декількома серверами. Такий підхід дозволяє значно знизити навантаження на сервер через те, що відбувається розподілення операцій, але в той же час цей підхід не є таким надійним, як двохрівнева архітектура.

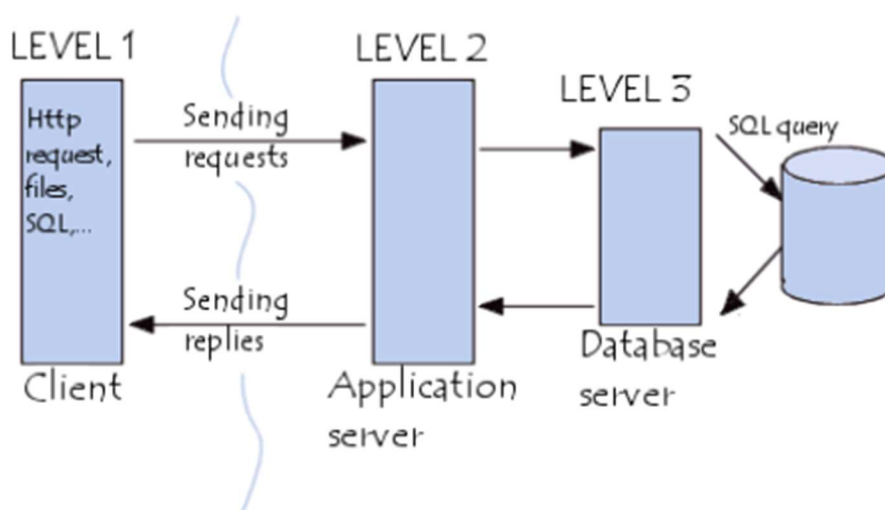


Рисунок 2.3 – Багаторівнева клієнт-серверна архітектура

Першим рівнем є клієнт, котрий дозволяє писати запити до БД, другий рівень – це програмне забезпечення СУБД, котрий інтерпретує запити і реалізує взаємодію між клієнтом та файловою системою. Третій рівень – безпосереднє сховище даних.

Перевагою моделі взаємодії клієнт-сервер є розподіл програмного коду на окремі частини, а саме – коду клієнтського застосунку та серверного, а також занижені системні вимоги для обчислювальних машин клієнтів, оскільки більшість операцій буде відбуватися на серверній стороні. До недоліків можна віднести вартість якісного мережевого обладнання, а також необхідність у кваліфікованому адміністраторі.

## 2.2 Сокети

Існує дуже багато застосунків на різних ОС, що взаємодіють між собою за допомогою мережевого з'єднання. Як приклад можна навести будь-який месенджер, такі як WhatsUp та Viber. Як правило, з'єднання між застосунками досягається шляхом використання сокетів.

Сокети – це інтерфейс, який дозволяє зв'язувати між собою різні пристрої, що знаходяться в одній мережі. Сокети існують двох типів: клієнт та сервер. Різниця між ними полягає у тому, що сервер прослуховує вхідні з'єднання і оброблює запити, що надходять, а клієнт до цього серверу підключається. Коли сервер запущений, він починає прослуховувати заданий порт на наявність вхідних з'єднань. Клієнт при підключенні має знати IP-адресу сервера та порт. В зв'язку з цим одним з основних практичних застосувань сокетів є їх використання в якості засобу для комунікації.

WebSocket - протокол напівдуплексного зв'язку над TCP-з'єднанням, призначений для обміну повідомленнями між клієнтом і веб-сервером в реальному часі. В даний час в W3C здійснюється стандартизація API Web Sockets. Чорновий варіант стандарту цього протоколу затверджений IETF.

WebSocket буде корисний розробникам, які створюють додатки з інтенсивним обміном даними, вимогливі до швидкості обміну і стабільності каналу. Наведемо приклади додатків, які використовують сокет для реалізації своєї бізнес-логіки. Котирування валют, акцій, біржова статистика - це все моніторинг в реальному часі, дані на клієнтів постійно оновлюються за допомогою сокет-з'єднання. У текстових чатах, незалежно від кількості користувачів, всі учасники бесіди практично одночасно отримують повідомлення один від одного за одним і тим же з'єднанням. Букінг (замовлення або бронювання чого-небудь) передбачає, що на запит користувача хтось повинен оперативно відгукнутися (наприклад, водій таксі), або сервер повинен забронювати для користувача послугу (наприклад, місце в кінозалі),

сповістивши при цьому його і інших користувачів про те, що дана послуга була заброньована. Головна вимога, яке виконує тут сокет-з'єднання - повний двосторонній зв'язок (повнодуплексне з'єднання), сервер в будь-який момент може звернутися до клієнта, а той йому відповісти, і навпаки.

### 2.3 Огляд Firebase

Керування мережевим обладнанням та хостингом не є простою задачею, і дедалі популярнішими стають так звані BAAS.

Backend as-a-Service став популярним рішенням для хмарних обчислень для розробників та підприємств, які не мають достатнього фінансування на створення власної, або технічної підтримки існуючої, інфраструктури. Як у багатьох сферах технологій, одну з провідних позицій на ринку BAAS займає Google Firebase.

Firebase – це платформа для розробки програмного забезпечення, що була запущена у 2011 році і придбана Google в 2014 році. Розроблена як база даних в реальному часі, зараз платформа налічує 18 повнофункціональних сервісів та спеціальні API. Вся платформа – це рішення Backend as-a-Service як для мобільних, так і для веб-додатків, що включає послуги зі створення, тестування, та керування додатками.

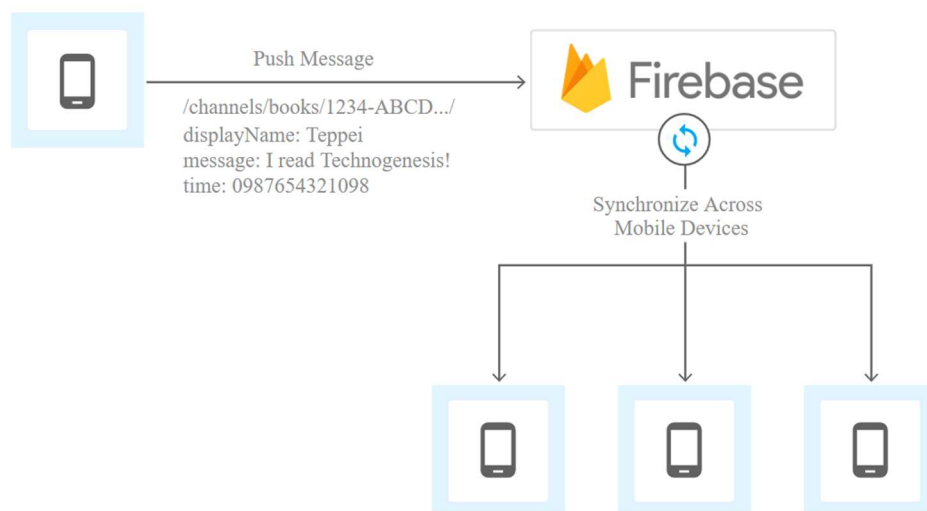


Рисунок 2.4 – Схема взаємодії з Firebase

Рішення BAAS дозволяють усунути необхідність в управлінні базами даних для резервного копіювання та отриманні відповідного мережевого обладнання. Натомість є можливість підключити сервіс до створеного додатку через спеціальні API для кожної окремої послуги. У Firebase є 7 з них, які охоплюють повний спектр бек-енд технологій для різноманітних додатків на різних платформах. Список платформ, з якими об'єднується Firebase, включає Android, IOS, Web, Unity.

Firebase має три категорії послуг, що можуть надаватись для програм клієнта. Вони поділяються на послуги для створення додатків, забезпечення якості додатків та інструменти для аналітики.

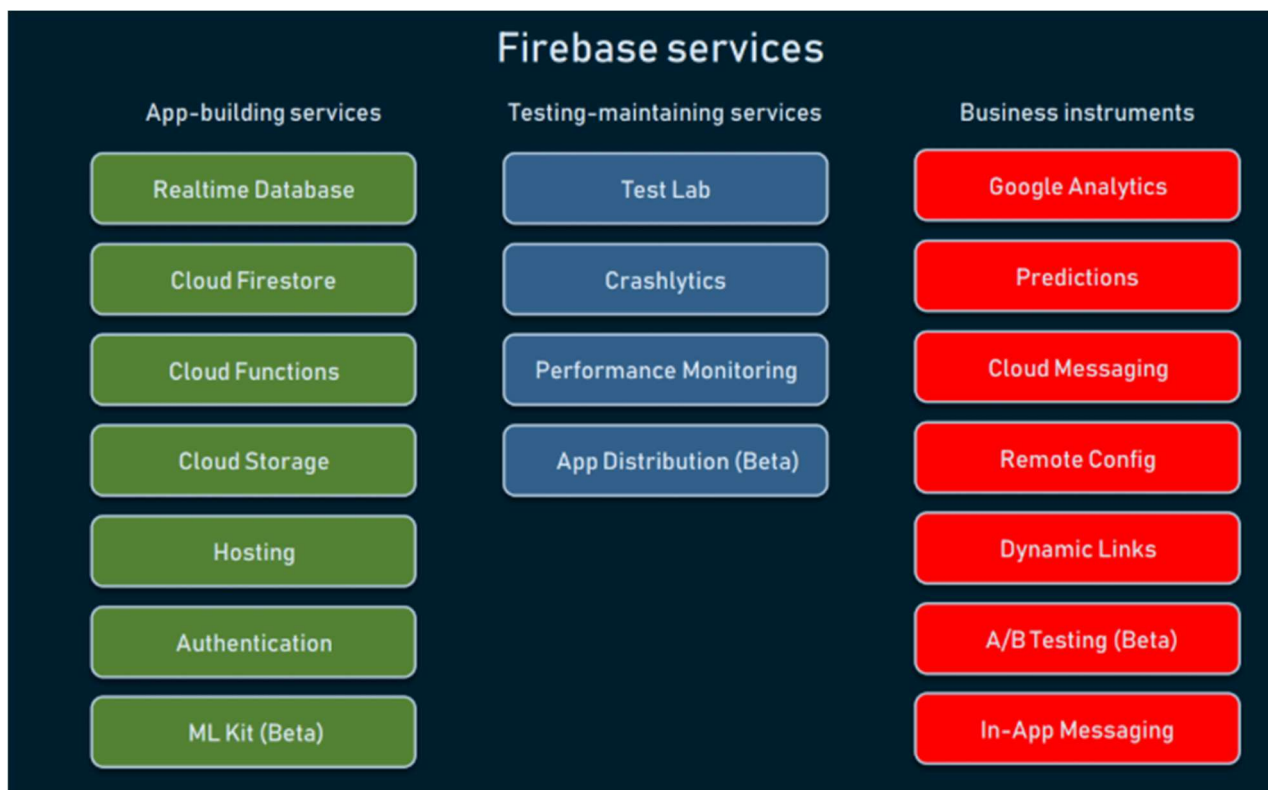


Рисунок 2.5 – Сервіси Firebase

Firebase Realtime Database була першим продуктом, який з'явився як продукт Firebase, тому це найвідоміший і стабільний сервіс на всій платформі. База даних в реальному часі - це по суті хмарне сховище NoSQL, яке можна підключити до програми, щоб забезпечити доступ у реальному часі до даних на різних платформах. Однією з переваг є те, що база даних може працювати в автономному режимі, кешуючи дані в пам'яті пристрою та, після підключення до Інтернету, синхронізуючи їх.

Дані зберігаються в JSON і користувачі можуть їх запитувати. З точки зору безпеки, база даних в реальному часі забезпечує доступ до даних на основі дозволів. Це можна зробити за допомогою аутентифікації Firebase та надання дозволів за особами користувача або правилами безпеки.



Рисунок 2.6 – Приклад представлення даних у БД

Cloud Firestore - це ще одна хмарна база даних NoSQL в реальному часі. На відміну від бази даних Firebase Realtime, Cloud Firestore розроблений для корпоративного використання, що тягне за собою масштабованість, складні моделі даних та розширені варіанти запитів. Консоль Firebase можна використовувати для перегляду даних в обох базах даних. Іншим спільним моментом є те, що існують SDK для роботи з кодом на стороні сервера обох баз даних. Вони доступні для Python, Node.js , Golang, Ruby, PHP, Java , .NET та C #.

Автентифікація Firebase - це функція аутентифікації Google, розроблена для програм, що використовують Firebase. Це дозволяє використовувати попередньо вбудований або створити інтерфейс користувача для аутентифікації користувача та увійти в систему користувачів через користувацькі облікові дані, електронні листи чи соціальні медіа.

Послуги хостингу . Якщо створюється прогресивний веб-додаток або мобільна цільова сторінка, беззаперечно знадобиться хостинг. Firebase пропонує статичний веб-хостинг для додатків, створених за допомогою HTML, CSS та JavaScript . З точки зору безпеки, він використовує стандартні протоколи HTTPS і SSL для доставки файлів та інших типів даних.

Хмарні функції . Це ще одна інтеграція існуючого продукту Google у Firebase. Це інструмент для запуску бек-коду з хмари на основі подій. Те, як Cloud Cloud функцій пропонує запустити програму, називається архітектурою без сервера . Цей тип архітектури означає побудову додатків у вигляді набору окремих функцій, ізольованих у хмарі та з'єднаних між собою за допомогою API. Цей тип архітектури додатків набував все більшої популярності кілька років тому, тому доречно бачити Google як одного з основних безсерверних провайдерів .

Test Lab - це послуга, яка надає віртуальний та фізичний пристрій для тестування додатка в реальному середовищі. Він може бути інтегрований до існуючого тестового середовища, наприклад Android Studio , або до інструментів тестування браузера. Тестова лабораторія надасть звіти про збої, скріншоти як результати тесту. Якщо відсутні тести, є бот, здатний сканувати програму, знаходити помилки та надавати звіти. Мінусом є те, що Test Lab не інтегрується з інструментами для тестування iOS.



Моніторинг продуктивності приносить автоматизацію відстеження ефективності програми в реальному часі. Цей інструмент надає інформацію про ефективність на основі першопричини, відстежуючи не тільки продуктивність програми, але і якість підключення до сервера та час відповіді для різних типів мережі.

Google Analytics займає перше місце, оскільки це відомий інструмент, інтегрований у платформу Firebase. Google Analytics надає цінні показники щодо збереження користувачів, показників активності користувачів чи будь-якого іншого типу даних поведінку користувачів. Що стосується Firebase, можливості звітності необмежені та безкоштовні.

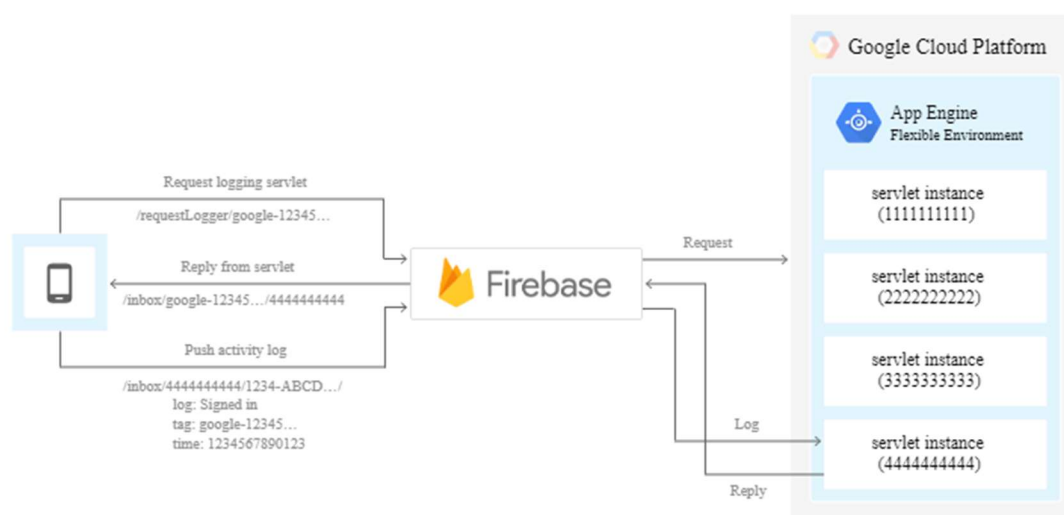


Рисунок 2.7- приклад використання Firebase

Необхідно розглянути переваги та недоліки Firebase, зосереджуючись на технічних перевагах платформи.

Можливості бази даних. Хоча це завжди залежить від бюджету, Google пропонує досить надійні бази даних для використання з програмами. Як Realtime, так і Firestore можна масштабувати за розміром, пропонуючи

повністю захищене кероване рішення, яке все ще забезпечує легкий доступ до даних через консоль Firebase. Оновлення даних та офлайн-доступ роблять бази даних для використання в реальному часі, а також підтримують синхронізацію декількох БД.

Широка кількість послуг. Firebase пропонує багато продуктів, щоб програма працювала. Є можливість вибрати між двома базами даних (база даних в реальному часі та Firestore), зберігати медіа у хмарі та будувати безсерверні програми за допомогою інтегрованих хмарних функцій.

Вільний старт. По-перше, Firebase не вимагає платежів за більшість своїх послуг і буде абсолютно безкоштовно. Це дозволить зрозуміти, чи відповідає він вашій заявці, і зрозуміти всі особливості. Як тільки отримується певний об'єм пам'яті бази даних або потрібна певна послуга, завжди можна вибрати між планами. Сторінка цін містить калькулятор цін, який можна регулювати за різними параметрами, оскільки це звичайна практика для хмарних сервісів.

Коротка документація. Хороша технічна документація, документація API, посилання на SDK - все, що робить будь-який продукт простішим у використанні та доступним для користувача. Досліджуючи сторінку продуктів Firebase, помітно, що вона містить всю необхідну інформацію щодо інтеграцій, доступних платформ, інструкцій та списків підтримуваних технологій. Крім того, є нагода перевірити канал YouTube Firebase, який здається досить активним з точки зору випуску нових відео та новин.

Доступний інтерфейс користувача та простота інтеграції. У більшості випадків Firebase вимагає мінімальних знань мови програмування та пропонує інтеграцію через свій користувацький інтерфейс.

Статичні можливості хостингу. Хостинг Firebase поставляється з мережею доставки вмісту (CDN), вбудованою в платформу Google Cloud. В основному, CDN - це мережа розподілених серверів, яка гарантує швидку доставку вмісту

					<b>ІАЛЦ.0454400.004 ПЗ</b>	<b>Лист</b> 25
<b>Зм</b>	<b>Лист</b>	<b>№ докум.</b>	<b>Підп.</b>	<b>Дата</b>		

по всьому світу. У випадку з Firebase, використання її статичного хостингу дозволяє легко створювати однотипні пейджери або веб-додатки.

Як і будь-яка технологія, firebase не позбавлена недоліків.

Обмеження бази даних у реальному часі Firebase.Здебільшого використовуватиметься база даних Realtime як основне сховище даних, яке має свої мінуси. Однією з головних проблем є обмежені можливості запиту. У базі даних в реальному часі неможливо відфільтрувати можливості, оскільки ціла БД - це величезний файл JSON, що ускладнює складання запитів.

Особливості, що стосується також БД Firebase Realtime та її моделювання даних. Через структуру «база даних як єдиний файл» неможливо реалізувати відношення між елементами даних.

Як результат, обидві ситуації можливі, коли мова йде про БД Firestore під час роботи на більш складних моделях даних. Тому переваги використання цієї системи є більш суттєвими.

Блокування провайдера.Проблема з блокуванням постачальника не співвідноситься з Firebase як платформою. У більшості випадків це проблема із використанням рішень BaaS взагалі. Однак, поки Firebase не надає жодних інструментів міграції для передачі ваших даних на іншу платформу, яку можна вважати підробкою.

Менша підтримка додатків для iOS.Хоча ми багато говоримо про його кросплатформенний характер, ви все ще можете помітити, що Firebase більше зосереджується на мобільній платформі Android (звичайно). Тестова лабораторія може бути легко інтегрована в студію Android і підтримує широкий спектр пристроїв Android для тестування. Щодо iOS, лише у 2018 році команда Test Lab оголосила про запуск закритої бета-підтримки для

пристроїв iOS; що включає базовий інтерфейс користувача та деякі тестові пристрої.

Враховуючи особливості Firebase, ви можете зрозуміти, чи добре вона підходить для вашої програми. Вибір постачальника - це завжди 50-відсоткове технологічне рішення, а інша половина припадає на репутацію та довіру постачальника. Отже, якщо ви відчуваєте, що Firebase нічого не пропонує для вашого точного проекту, ви можете подивитися на альтернативних постачальників на ринку.

Для взаємодії з Firebase використовуються наступні інструменти:

Консоль Firebase . Усіма проектами Firebase можна керувати з одного місця, консолі Firebase. Це уніфікована панель управління, на якій ви можете побачити всі ваші програми, пов'язані з послугами Firebase. Проекти виконують роль віртуальних контейнерів для коду програми та налаштувань сервера, що дублює Google Cloud Platform .

Firebase CLI . Використовуючи інтерфейс командного рядка Firebase можна взаємодіяти з даними, що зберігаються в базах даних, дійти до конфігурацій сервера та розгортати проекти Firebase.

Кількома роками раніше найкращим пристосуванням для Firebase вважали невелику програму, яка не потребує складних запитів даних і покладається на оновлення в реальному часі. Незважаючи на те, що вона все ще залишається основним напрямком роботи бази даних Firebase Realtime, з виходом Firestore вона стала доступною для корпоративних програм. Більше того, оскільки його SDK став відкритим кодом, немає реальної різниці між тим, що вважалося перевагою Parse перед Firebase.

Враховуючи всі фактори, можна проаналізувати пропозиції, що можуть запропонувати інші постачальники. Основними конкурентами на ринку є:

AWS Aurora . Основна перевага сервісу AWS в BaaS перед Firebase полягає в тому, що він використовує реляційну базу даних, яка гарантує кращу структуру даних та запити. Він більше розроблений для корпоративних програм та підтримує міграцію даних з платформи на платформу. Таким чином, цей пункт може переважати технічні особливості, оскільки не буде блокування постачальника.

Parse сервер . Хоча епоха платформи Parse минула в 2017 році, все ще доступна її версія з відкритим кодом. Серед її плюсів є багато функцій перед складанням, загальна настройка, вибір середовища для розміщення. З іншого боку, його інтерфейс командного рядка досить нестабільний, і його підтримує спільнота розробників, а не велика корпорація.

Back4App . Це реалізація Parse Server, яка в основному виправила свої недоліки. Back4App пропонує аналогічні функції, що виконує Firebase, за винятком, більш гнучкого моделювання даних та налаштувань запитів до бази даних.

## ВИСНОВОК ДО РОЗДІЛУ 2

В розділі був виконаний аналіз технологій для створення клієнт-серверних додатків, опис та принципи їх роботи, а також короткий опис обраного сервісу Firebase, його можливості та переваги в порівнянні з конкурентами.

					<b>ІАЛЦ.0454400.004 ПЗ</b>	Лист
						29
Зм	Лист	№ докум.	Підп.	Дата		

### 3. ТЕХНІЧНА РЕАЛІЗАЦІЯ СТУДЕНТСЬКОЇ СОЦІАЛЬНОЇ МЕРЕЖІ НА БАЗІ ОПЕРАЦІЙНОЇ СИСТЕМИ ANDROID

#### 3.1 Вибір платформи для розробки ПЗ

Тема вибору платформи для розробки веб додатків завжди була і буде актуальною. Незважаючи на це, достойною популярністю користуються обидва варіанта, кожен з яких має свої сильні та слабкі сторони.

Вибір платформи відбувається на основі декількох критеріїв.

Перш за все треба визначити мету, яку хоче досягти розробник в результаті, цільову аудиторію користувачів та бюджет.

Смартфони складають приблизно 80% пристроїв, які використовуються для пошуку в Інтернеті. Станом на сьогодні вони стали невід'ємною частиною життя, без якої сучасній людині буде досить незручно. Оскільки користувач завжди має безпосередній доступ до мобільного пристрою, то і швидкість реагування на термінові чи важливі події є значно швидшою. Оскільки застосунок орієнтований на використання сучасною молоддю, то вибір смартфонів під цільову платформу для розробки є досить доречним, адже клієнтське програмне забезпечення різноманітних месенджерів та соціальних мереж користується великим попитом саме для мобільних пристроїв.

Існує два різних способи технічної розробки і реалізації проєктів на мобільних пристроях, а саме мобільні веб-сайти та мобільні застосунки. На перший погляд ці способи виглядають дуже схожими, проте мають свої переваги і недоліки. Мобільний сайт. Працює виключно через браузер. Відмінність від звичайних веб-сайтів полягає в тому, що вони розроблені з врахуванням адаптивної верстки, що забезпечує правильне відображення на будь-яких дисплеях. Серед переваг можна виділити:

- Сумінсість. Зручність роботи на різних типах смартфонів та планшетних ПК. Не потребує окремої версії з врахуванням особливостей різних ОС. Підтримка легкої інтеграції з функціями для сканування QR кодів та текстових повідомлень.
- Охоплення більш широкої цільової аудиторії. Завдяки підтримці багатьох пристроїв, яку забезпечує адаптивний веб-дизайн, задіяна більш широка аудиторія користувачів
- Підтримка та обслуговування. В порівнянні з застосунком, котрий потребує завантаження кожного оновлення, адаптивні веб-сайти дозволяють гнучко змінювати конфігурацію.

Існують також і недоліки:

- Зручність. На відміну від застосунків, адаптивні веб-сайти не можуть ефективно використовувати всі функції смартфона. Камери, GPS та інші функції, інтегровані в пристрої, не завжди зручно використовувати для адаптивних веб-сайтів.
- Розмір екрану пристрою. Через невеликі розміри екрани смартфонів / планшетів відображають набагато менше контенту в порівнянні з монітором настільного ПК або екраном ноутбука. Незважаючи на те, що адаптивний веб-дизайн динамічно підлаштовується під потрібний розмір екрану, він фактично зменшує і перебудовує контент, доступний на робочому столі.
  - Автономний доступ. Автономний режим функціонування мобільного сайту можливий тільки при використанні кешованих сторінок. Повноцінна робота вимагає гарного підключення до Інтернету.

На відміну від адаптивних / мобільних веб-сайтів, що працюють через браузер, нативні додатки повинні бути завантажені з певних порталів, таких як Google Play Market, App Store або інших.

Зм	Лист	№ докум.	Підп.	Дата

**ІАЛЦ.0454400.004 ПЗ**

Лист  
31



Мобільні програми розробляються окремо для кожної операційної системи, вимагають установки, забезпечують більш швидкий доступ до вмісту. Мають наступні переваги:

- **Зручність.** Аналіз показує, що додатки більш популярні, ніж аналогічні веб-сайти, оскільки більш зручні. Вони забезпечують кращу взаємодію з користувачем, швидше завантажують контент, простіше у використанні. Крім цього, мають push-повідомлення та дизайн, який більш гнучко сумісний з різними розмірами екрану.
- **Персоналізація.** Мобільні програми є відмінним рішенням для служб, які вимагають регулярного використання. Вони дозволяють користувачам створювати особисті облікові записи, а також зберігати важливу інформацію під рукою.
- **Робота в автономному режимі.** Оскільки додатки вимагають установки, вони можуть надавати доступ до своїх функцій і контенту навіть без підключення до Інтернету.

Недоліки існують наступні:

- **Сумісність.** Забезпечення належного функціонування нативного додатку залежить від вимог конкретної операційної системи. Це означає, що для кожної платформи (iOS, Android, Windows) потрібна окрема робоча версія програми.
- **Підтримка, обслуговування .** Коли додаток розробляється для кількох різних платформ, його підтримка вимагає більше часу і грошей. Необхідно регулярно надавати оновлення, виправляти проблеми сумісності з кожним типом пристроїв. Крім того, потрібно завжди нагадувати користувачам про необхідність установки нових оновлень.

Підводячи підсумок, незважаючи на плюси адаптивного веб-дизайну, мобільні додатки є більш зручними у практичному використанні. Вони використовуються у наступних випадках:

- Для вирішення конкретних бізнес-потреб. Наприклад, створення мобільних версій соціальних мереж забезпечує їх динамічне функціонування, постійний доступ користувачів до збереженого контенту (зображень, тексту, музики, відео).
- Додатки ідеально підходять для інтерактивних ігор (Angry Birds), інструментів повсякденного використання (Evernote), сервісів для тренувань з відстеженням процесу в реальному часі і можливістю відправки регулярних нагадувань (FitBit, Duolingo).
- Крім того, нативні додатки є гарним рішенням для маніпулювання даними. Наприклад, для банківських операцій і послуг мобільна версія може стати відмінним доповненням до повноцінного веб-сайту.

### 3.2 Життєвий цикл розробки ПЗ



Рисунок 3.1 – Схема життєвого циклу розробки ПЗ

Частковий опис життєвого циклу розробки додатку орієнтованого на мобільні платформи, а точніше – під ОС Android.

Аналіз ідей та збір вимог. Головна ідея цього етапу полягає у формуванні бачення готового продукту, аналіз потрібного та не потрібного функціоналу. На цьому етапі формується ТЗ. Для переходу на наступний етап потрібно мати сформоване бачення результуючого продукту а також технічну документацію-опис майбутньої розробки. Головні проблеми, що виникають через пропуск цього етапу, пов'язані з витратою часу на переписування коду проєкту а також мати уявлення, коли продукт буде в стані, готовому до релізу.

Прототипування мобільного додатку. Ціль даного етапу – швидко та просто візуалізувати додаток для подальшої розробки дизайну.

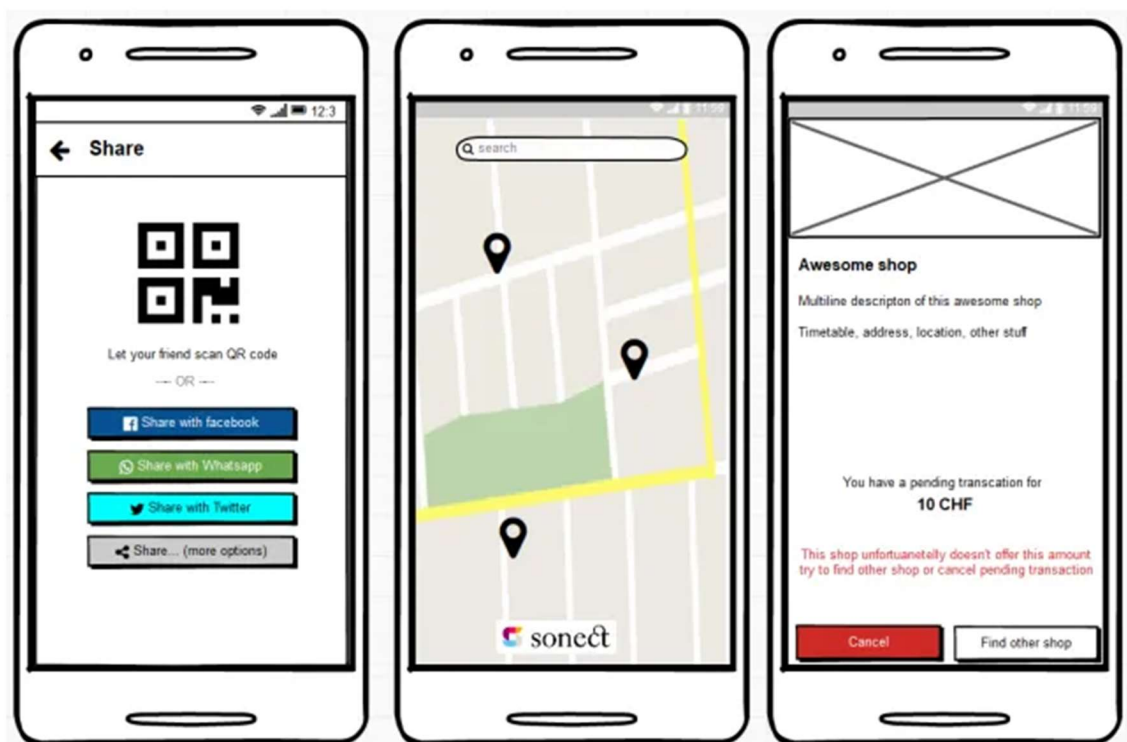


Рисунок 3.2 – Прототипування мобільного додатку

Коли готові макети окремих екранів необхідно створити мапу переходів між ними. Для подібної роботи існує багато готових сервісів.

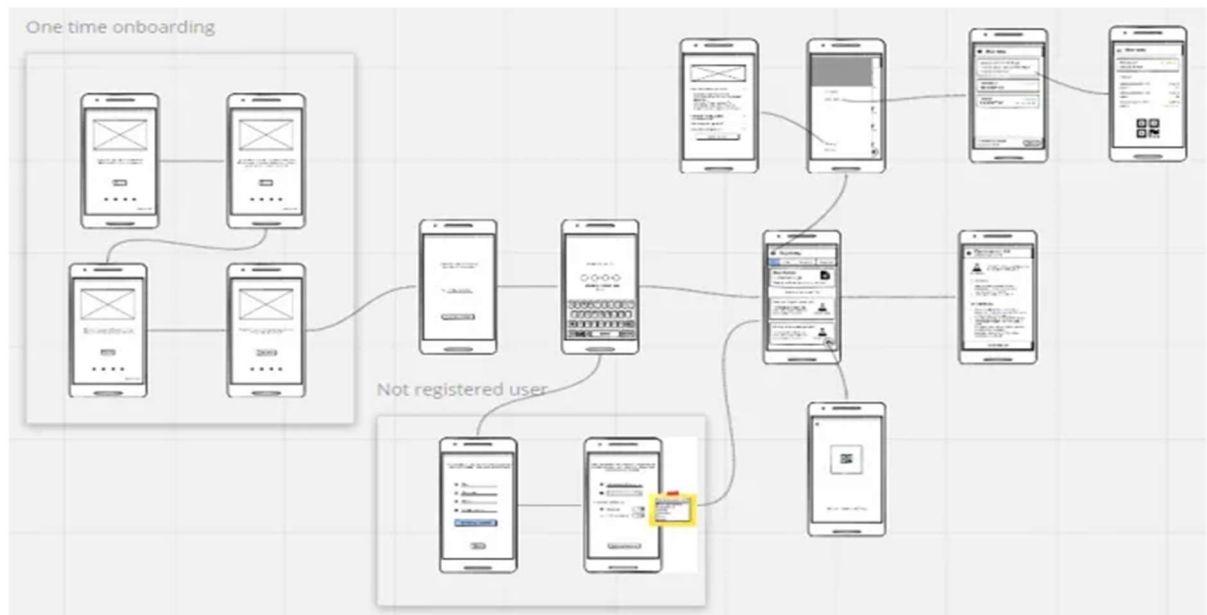


Рисунок 3.3 – Мапа переходів між екранами програми

Дизайн мобільного додатку. На цьому етапі створюється безпосередній дизайн для екранів застосунку. Основний інструмент – Sketch, Zelpin.

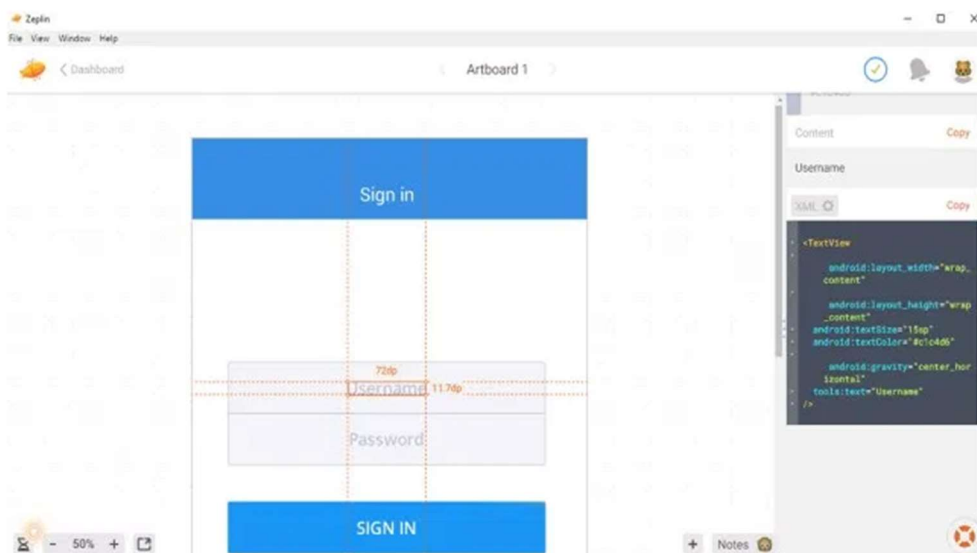


Рисунок 3.4 – Приклад дизайну представлення

Для анімованих елементів доцільно використовувати Lottie.

Розробка. На цьому етапі виконується безпосередня розробка майбутнього додатку.

Деплой. Цей етап передбачає публікацію готового продукту на спеціальній платформі Google play.

Тестування та підтримка розробленого додатку. Етап супроводження створеного додатку.

### 3.3 Вибір інструментарію

Для програмування застосунків для ОС Android існує дві основні мови програмування – Java та Kotlin. При створенні даного дипломного проекту було використано мову програмування Java. Серед переваг цієї мови програмування можна виділити такі: мова є об'єктно-орієнтованою, простий синтаксис, розроблена з урахуванням незалежності від платформи, на якій виконуватиметься код, адже використовується віртуальна машина.

В якості IDE використовувалось спеціальне ПЗ – Android Studio.

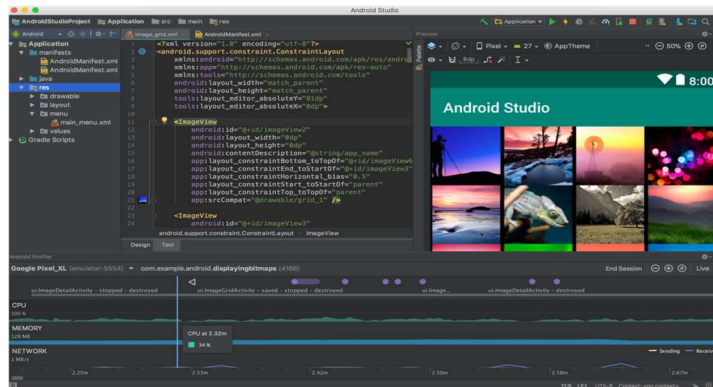


Рисунок 3.5 – Середовище Android Studio

Android Studio – інтегроване середовище розробки для роботи з платформою Android. Воно знаходиться у вільному доступі, засноване на програмному забезпеченні IntelliJ IDEA від компанії JetBrains. Цей додаток має репутацію дуже зручного та надійного у використанні. Серед основних переваг та функцій можна виділити наступні:

- середовище розробки підтримує роботу з декількома мовами програмування, до яких відносяться найпопулярніші - C / C ++, Java.
- редактор коду, з яким зручно працювати;
- дозволяє розробляти програми не тільки для смартфонів / планшетів, а й для портативних ПК, приставок для телевізорів Android TV, пристроїв Android Wear, новомодних мобільних пристроїв з незвичайним співвідношенням сторін екрану;
- тестування коректності роботи нових ігор, утиліт, їх продуктивності на тій чи іншій системі, відбувається безпосередньо в емуляторі;
- рефакторинг вже готового коду;
- досить велика бібліотека з готовими шаблонами і компонентами для розробки ПО;
- розробка програми для Android N - самої останньої версії операційної системи;

Зм	Лист	№ докум.	Підп.	Дата

**ІАЛЦ.0454400.004 ПЗ**

Лист  
37

- попередня перевірка вже створеного додатку на предмет помилок в ньому;
- великий набір засобів інструментів для тестування кожного елемента програми, ігри;
- для недосвідчених / початківців розробників спеціально створено посібник з використання Android Studio, розміщене на офіційному сайті утиліти.

Незважаючи на наявність вбудованого Android-емулятора в самому середовищі розробки, з тестуванням новорозробтанного додатку можуть виникнути труднощі. Так, для його запуску необхідна досить значна за продуктивністю апаратна основа ПК, на якому планується тестування.

Ще один недолік - це неможливість написати серверні проекти на мові Java для ПК, Android пристроїв.

### 3.4 Опис структури додатку

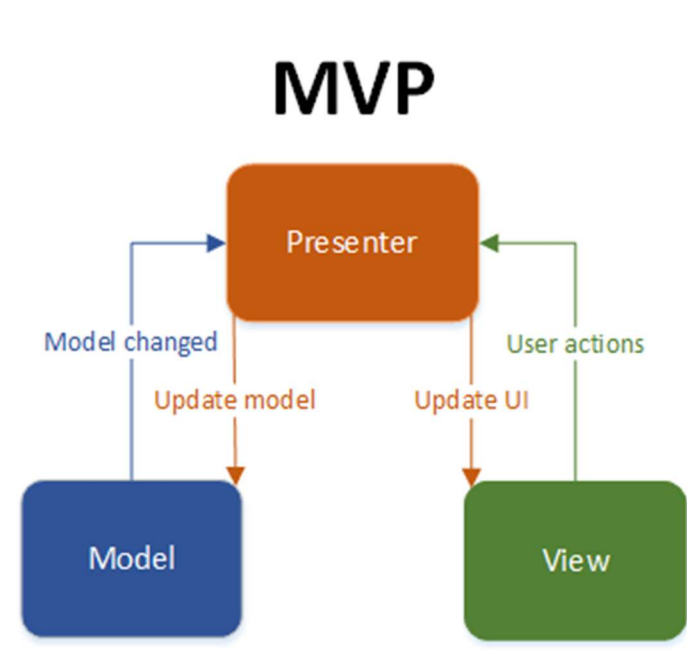


Рисунок 3.6 – Схема патерну MVP

Додаток розроблено за класичним підходом MVP. Суть цього патерну полягає у розподілі додатка на три компонента: модель, представлення, пред'явник.

Необхідність у цьому шаблоні виникає при наступних ситуаціях: виділення

бізнес-логіки від логіки інтерфейсу користувача, щоб зробити його простішим для розуміння та підтримки, надання доступу до коду різним сторінкам, що потребують однакової поведінки. Розглянемо складові частини патерну. Модель працює з даними, проводить обчислення та керує усіма бізнес-процесами. Представлення показує користувачеві інтерфейс та данні з моделі. Пред'явник слугує проміжною ланкою між моделлю та представленням. Розглянемо схему роботи застосунка за патерном MVP:

- 1) Представлення будує інтерфейс і додає в нього дані з моделі.
- 2) Користувач бачить інформацію та взаємодіє з інтерфейсом.
- 3) Представлення перехоплює події та делегує їх пред'явеику.
- 4) Пред'явник оброблює дані та передає їх моделі.
- 5) Модель виконує необхідні обчислення та дії.
- 6) Представник отримує оновлену модель та передає її представленню.
- 7) Представлення будує інтерфейс з новими даними.

Патерн слугує для спрощення розробки графічних інтерфейсів і широко використовується при проектуванні та розробці Android додатків.



## ВИСНОВОК ДО РОЗДІЛУ 3

В розділі були виконані наступні завдання:

1. Обґрунтування вибору платформи для розробки додатку
2. Проаналізовані принципи, за якими ведеться розробка додатків, що орієнтовані на використання для мобільних платформ
3. Опис структури програми

Для використання було запропоноване оптимальне структурне рішення для створення додатку за темою дипломного проєкту. Обрані технології допоможуть створювати програмний продукт більш швидко та зручно, а засоби для тестування дають можливість досконало перевірити роботу нової розробки.

					<b>ІАЛЦ.0454400.004 ПЗ</b>	Лист
						40
Зм	Лист	№ докум.	Підп.	Дата		

## 4. ОПИС РОЗРОБЛЕНОГО ДОДАТКУ

### 4.1 Опис інтерфейсу додатку

Більш детальний огляд інтерфейс застосунку, який був розроблений у цьому дипломному проєкті. При відкритті додатку вперше, користувач потрапляє на форму реєстрації.

Hello there,  
welcome back

Email

Password

[Forgot your Password ?](#)

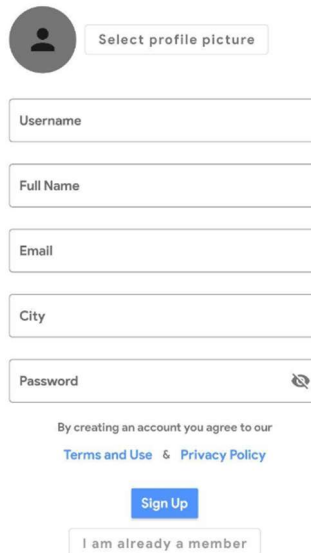
[Sign In](#)

[New here ? Sign Up instead](#)

Рисунок 4.1 – Форма входу до системи

Для входу в систему необхідно ввести свою електронну пошту та пароль. Якщо користувач вперше користується додатком, йому пропонується виконати процес реєстрації нового користувача мережі. Для цього користувач буде переміщений на форму реєстрації.

## Get on Board



Registration form titled "Get on Board". It includes a profile picture selection button, and input fields for Username, Full Name, Email, City, and Password. Below the fields, there is a note: "By creating an account you agree to our [Terms and Use](#) & [Privacy Policy](#)". At the bottom, there are two buttons: "Sign Up" (blue) and "I am already a member" (grey).

Рисунок 4.2 – Форма реєстрації у системі

Після заповнення форми реєстрації і відправки даних, пропонується виконати підтвердження вашого акаунта за допомогою електронної пошти, засобами Firebase Authentication.

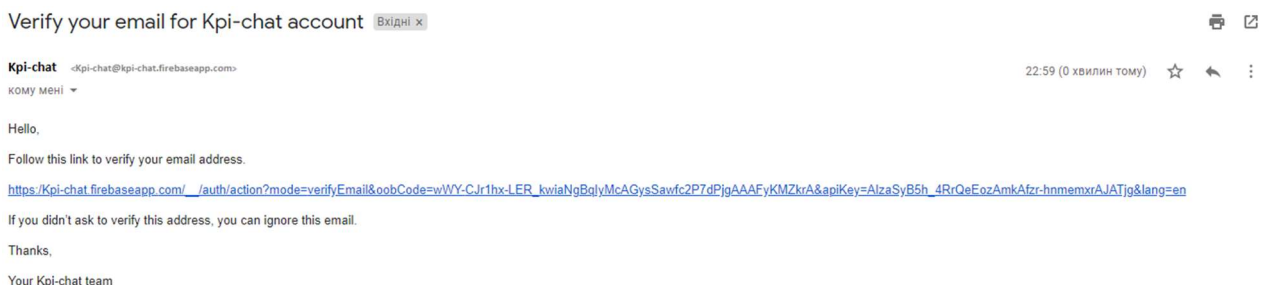


Рисунок 4.3 – Приклад згенерованого листа для активації облікового запису

Після успішного входу до системи, користувач потрапляє на головний екран, який виконує роль головної інформаційної стрічки мережі. В даний момент тут відображаються наукові статті.

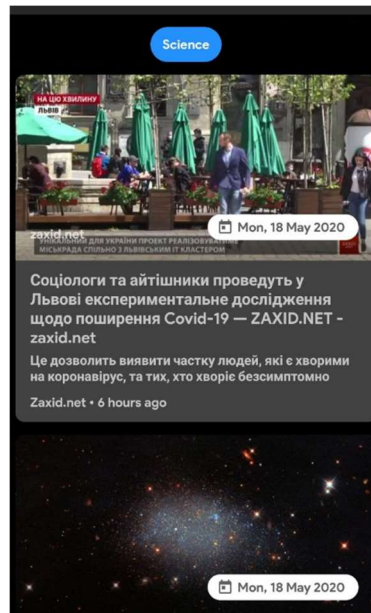


Рисунок 4.4 – Інформаційна стрічка

Навігація здійснюється за допомогою висувного меню.

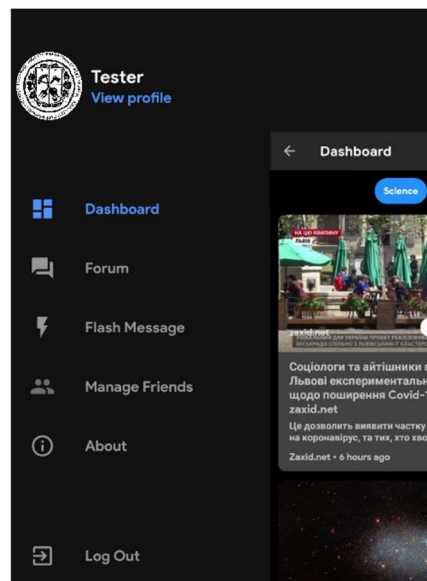


Рисунок 4.5 – Навігаційне меню

Для здійснення обговорень між групою користувачів створено спеціальний форум, що передбачає створення тегів для кожного запису, для більш швидкого та зручного пошуку потрібного розділу.

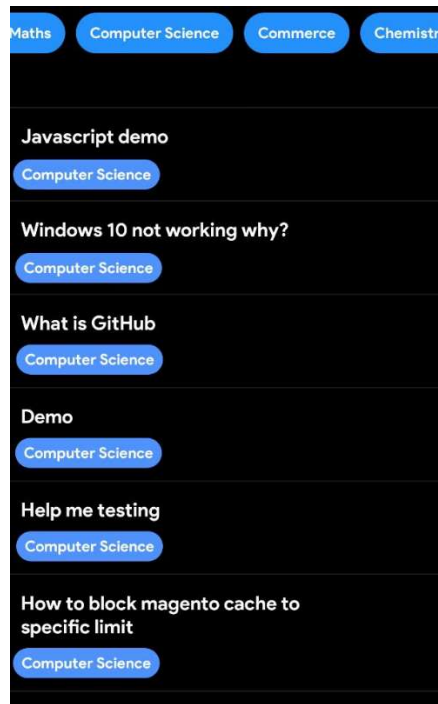


Рисунок 4.6 - Форум

Додаток передбачає створення сповіщень, редагування особистої сторінки, пошук друзів (контактів).

#### 4.2 Рекомендації щодо покращення додатку

Оскільки додаток розроблений за патерном MVP, це передбачає полегшену інтеграцію змін та покращень. Однією з основних рекомендацій для подальшої розробки є визначення з додатковим функціоналом CRM систем, що можуть бути корисними при організації навчального процесу та їх подальша інтеграція у додаток. Це зробить додаток більш самостійним та гнучким у використанні для всіх цілей, притаманних корпоративним застосункам. Також можна рекомендувати в подальшій розробці приділити увагу реалізації додатка на всіх популярних платформах, що має збільшити кількість користувачів, адже деякі операції виконувати зручніше за звичними ПК.

## ВИСНОВОК ДО РОЗДІЛУ 4

Було продемонстровано користувацький інтерфейс розробленого програмного продукту, опис базових можливостей, демонстрація роботи та вказівки для коректного використання.

					<b><i>ІАЛЦ.0454400.004 ПЗ</i></b>	<b><i>Лист</i></b>
<b><i>Зм</i></b>	<b><i>Лист</i></b>	<b><i>№ докум.</i></b>	<b><i>Підп.</i></b>	<b><i>Дата</i></b>		<b><i>45</i></b>

## ВИСНОВОК

Головною метою даного дипломного проекту є створення зручного та самостійного додатку, що здатний значно покращити внутрішню комунікацію в навчальних установах. Застосунок створено з використанням сучасних технологій та патернів проектування, що забезпечує додатку легку модифікацію та подальшу розробку. Основною задачею цього ПЗ є заміна більш не практичних засобів обміну повідомленнями та не спеціалізованих застосунків, котрі не передбачають їх використання з робочою метою.

У додатку реалізовані ключові для даного виду програм функції, котрі допоможуть користувачами отримати новий та універсальний інструмент здатний вирішити буденні проблеми при організації навчального процесу поза аудиторіями.

Загалом розроблена система продемонструвала свою надійність та здатність до існування. Також програмний продукт передбачає багато різноманітних нових рішень щодо покращення. Можливість реалізації нового функціоналу а також безпроблемна інтеграція з існуючою системою, завдяки використаним шаблонам розробки програмного забезпечення, дозволяє подовжити термін служби розробки на декілька років вперед.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Mark Murphy – “The Busy Coder`s Guide to Advanced Android development”, CommonsWare (2009).
2. Bill Philips - “Android Programming: The Big Nerd Ranch Guide”, Big Nerd Ranch Guides, 2013.
3. Alex Backman – “How to deliver client-server application that work”, 1995.
4. Robert Kermel – “Implementing Application Solutions in a Client-Server Environment”, 1995.
5. Casimir Saternos – “Client-Server Web Apps with JavaScript and Java”(2014).
6. Robin Wieruch – “The Road to React with Firebase” (2019).
7. Ashok Kumar S – “Mastering Firebase for Android Development: Build Real-time, Scalable Apps” (2018).
8. Laurence Moroney – “The Definitive Guide to Firebase: Build Android Apps on Google`s Mobile Platform” (2017).
9. Ted Hagos – “Learn Android Studio 3 with Kotlin: Efficient Android Development” (2018).
10. Android Documentation – [Електронний ресурс] – Режим доступу : [developer.android.com/docs](https://developer.android.com/docs).
11. Firebase Documentation – [Електронний ресурс] – Режим доступу : [firebase.google.com/docs](https://firebase.google.com/docs).
12. Android Studio for beginners, Part 1: Installation and setup [сайт]. URL: <https://www.javaworld.com/article/3095406/android/android-studio-for-beginners-part-1-installation-and-setup.html>
13. Model-View-Controller [сайт]. URL: <https://ru.wikipedia.org/wiki/Model-View-Controller>



14. Упростите вход с помощью быстрой и надежной аутентификации Firebas [сайт]. URL: <https://developer.android.com/distribute/best-practices/develop/firebase-authentication?hl=RU>
15. Различия между MVVM и остальными MV\*-паттернами [сайт]. URL: <https://habr.com/ru/company/mobileup/blog/313538/>
16. Developers [сайт]. URL: <https://developer.android.com/>